Timer Controller

Table of Contents

[Overview 2](#_Toc353172137)

[Registers 2](#_Toc353172138)

[TIMERCTRL0..3 2](#_Toc353172139)

[TIMERCOUNT0..3 2](#_Toc353172140)

[Examples 3](#_Toc353172141)

# Overview

The VideoChip offers 4 general purpose timers. Each timer can generate an interrupt. Timers can run at a variety of periods. They could be used, for example, to generate appropriate timing for a MOD (music) player, or periodic game logic.

All timers are simple count-up timers. When their count reaches the specified value, they set a flag indicating that they have expired, and then automatically reset their count to 0 and resume counting again.

You can read the current count, as well.

# Registers

All registers come in 4 sets each, since there are 4 timers available.

## TIMERCTRL0..3

This register allows for the configuration of a timer

76543210  
ExxxxxxR

E: 1 = Timer is Enabled. NOTE: This can disable the timer’s counter, but does NOT disable the  
 timer’s interrupt flag. You must do that separately.  
R: 1 = Repeat (auto-reset to 0). 0 = Oneshot (auto-disable at count).

*NOTE: The reason for choosing such a low divider (64), which yields a largely unusable frequency range (Z80 cannot take advantage of it very well), is to provide a high quality of granularity for choosing the specific frequency that you’re after. Eg. 50Hz or 60Hz, for music (MODs). Note that the VBlank on this computer occurs at 70Hz, meaning that exiting MOD files would not sound right if timed from the VBlank.*

## TIMERPRESCALE0..3

Each timer has a prescale value, which is adjustable from /64 to /65344. It is prescaling the source frequency of 62Mhz.

111111  
5432109876543210  
VVVVVVVV01000000  
0000000001000000 64 (good for 50Hz, 60Hz, etc)  
1111111101000000 65344 (just over 1 minute delay)

## TIMERCOUNT0..3

This 16bit register controls the count at which the timer will restart. Reading from this register safely, involves reading first the low byte then the high byte. Writing to this register safely, involves writing first the low byte then the high byte. The timer immediately resets to 0 when the high byte is written.

This counter is counting the output frequency of the Prescaler.

# Examples

Frequency TIMERPRESCALE TIMERCOUNT Error  
10Hz /64 96875 0%  
50Hz /64 19375 0%  
60Hz /64 16146 0.001% error  
1Hz /65344 949 0.018% error